

1994020884

N94- 25366

CONSIDERATIONS REGARDING THE DEPLOYMENT OF HYPERMEDIA AT JSC

**Final Report
NASA/ASEE Summer Faculty Fellowship Program - 1993
Johnson Space Center**

Prepared By: Charles J. Kacmar, Ph.D.

Academic Rank: Assistant Professor

University & Department: Florida State University
Department of Computer Science
Tallahassee, Florida 32306

NASA/JSC

Directorate: Engineering

Division Automation and Robotics

Branch: Intelligent Systems

JSC Colleague: Jack Bacon

Date Submitted: July 30, 1993

Contract Number: NGT-44-001-800

ABSTRACT

Electronic documents and systems are becoming the primary means of managing information for ground and space operations at NASA. These documents will utilize hypertext and hypermedia technologies to aid users in structuring and accessing information. Documents will be composed of static and dynamic data with static data consisting of traditional objects such as text, graphics, audio and video and dynamic data consisting of user-defined annotations and hypermedia links.

The report consists of three major sections. First, it provides an overview of hypermedia and surveys the use of hypermedia throughout JSC. Second, it briefly describes a prototypical hypermedia system that was developed in conjunction with this work. This system was constructed to demonstrate various hypermedia features and to serve as a platform for supporting the electronic documentation needs for the MIDAS system developed by the Intelligent Systems Branch of the Automation and Robotics Division [Pac92]. Third, it discusses emerging hypermedia technologies which have either been untapped by vendors or present significant challenges to the Agency.

INTRODUCTION

The purpose of this document is to provide a brief overview and assessment of hypermedia¹ use at the Johnson Space Center. Due to the variety of applications which utilize, behave like, or plan to incorporate hypermedia, it is not the purpose of this document to argue a specific set of requirements. However, this document will discuss some of the primary and advanced hypermedia capabilities that relate to ongoing work at JSC.

The concepts in this document are a result of many interactions with groups throughout JSC. The author worked closely with staff of and contributed specifically to the Electronic Documentation (EDP) and HYPERMAN projects. Although many systems were viewed and a prototype was constructed, this report will not focus on any particular hypermedia model or system.

Organization

This report consists of three major sections. The first section will present an overview of hypermedia and establish terminology which will be used throughout the remainder of the report. Included in this section are small scenarios and discussions of capabilities that are characteristic of hypermedia systems. Section two presents a very brief overview of a prototype hypermedia system which demonstrates hypermedia features and provides a platform for supporting electronic documentation. Section three discusses features which are important to JSC but do not appear to be adequately addressed by vendors. Thus, systems which lack these features may not be sufficient to meet some of the current and especially the future needs of the Center. The lack of support for these features also indicates areas for future research and further investigation.

BACKGROUND

Hypermedia

Hypermedia is a methodology for storing and accessing information by association. It allows relationships to be defined among information objects to provide access to those objects by navigation.

For the purpose of this document we define the basic components of hypermedia to be *nodes*, *objects*, *anchors*, *links*, and *link markers*². Nodes are objects which contain other objects, i.e. a container, including other nodes. An object is an entity which has content or substance such as a text string with attributes and values. An object may be composed

¹ The terms *hypertext* and *hypermedia* will be regarded as synonymous in this document [Nel65].

² Actual terminology used at JSC for these entities is slightly different. A *link* is an action that associates two objects. There are two basic types of links: a *traversal link* which causes the user to traverse to another location, and an *action link* which links to an action. A *markup* is a visible modification to the text on a page of a document [which includes links]. The term *hyper-link* means: a markup object which connects to another markup or to an action. An *icon anchor* is a markup represented by an icon. Even though differences exist in terminology, it is assumed that the reader can relate the two sets of terms [NAS93].

of other objects. An anchor relates nodes and objects to links. A link relates anchors.

The anchor-link-anchor relationship exists for the purpose of defining associations among objects in the hypermedia information space. An association is defined, informally, as a tuple consisting of

(nodes:objects, Anchors, Links, Anchors, nodes:objects)

In other words, a node, object, or collection of nodes and objects are related by the existence of anchors and links. A link may connect one or more nodes and objects at the source end of the link to one or more nodes and objects at the destination end of the link. An object or node may be associated with multiple anchors and links.

The visual manifest of a link on the user display is called a *link marker*. Link markers may appear in different ways such as icon, enclosed text string, a text string in a special font, or as an object which is displayed in a different color. Although links may exist without the need for link markers, in most situations, a link marker will appear at the source end of the link.

A link is *uni-directional* if it is navigable only from the source to the destination. A link which is navigable in both directions, from the source to the destination and from the destination to the source, is said to be *bi-directional*. Bi-directional links are normally accompanied by link markers which are visible at both ends of the link. Systems which provide both types of link directionality are preferred.

To be consistent with the literature, we allow the possibility of links connecting to other links thus forming a "strap"-like relationship. Navigating from the source of a strap-link to the strap will cause the ends of the strap to be activated thus providing a 1-to-N type of relationship. Although links are normally viewed and used to define 1-to-1 relationships, the implementation of links may provide M-to-N relationships in either their basic form or through the use of strap links.

Nodes, objects, anchors, links, and link markers are all assumed to be accompanied by appropriate functionality to support their existence, persistence, and operation. This functionality may reside in one or more components of the application - the application's interface, the body of the application, or the application's data management component. The placement of functionality is arbitrary and is beyond the scope of this report.

Granularity

Many organizations, including JSC, require the hypermedia system to operate in parallel with a paper document environment. The hypermedia system must mimic paper documents by presenting information in page-sized chunks. Though this strategy is effective for certain documents, a paper-based metaphor for accessing and managing electronic documents may in fact be counterproductive to user activity since the "page" may be too small or too large for certain information. The electronic environment increases the flexibility of presentation by allowing the *granularity* of information which is displayed to a user to be changed. Thus, information is viewed in terms of "chunks" rather than pages. The quantity of material in a chunk is variable and may depend upon

the complexity of the material, the display surface on which it is viewed, and level of user sophistication.

Another issue regarding granularity concerns the size of objects which may be associated with a link. Some hypermedia systems provide an object-like model in which the size of an object is determined by the user. If the user creates an object which is the size of a paragraph, the smallest identifiable unit of the object is the entire paragraph. Systems which provide character and pixel-level support may seem to provide greater flexibility but this is questionable due to the manner by which the information is used and amount of information which the user needs to associate with links. Of more importance to NASA is the ability to define link endpoints consisting of words or phrases which are embedded within textual information as well as portions of graphical objects. The critical issues here concern the association of link endpoints with their intended information and the maintenance of those relationships across subsequent versions of the document.

Annotations

Annotations provide a mechanism for personalizing documents with textual comments, graphical objects, drawings, and user-defined links. The two common methods of supporting annotations are popup notebbox and marginalia (writing in the margins).

A popup notebbox is requested by the user into which comments are placed. The notebbox can be connected or associated with the original document in several ways but is usually done so with a link. The tool which supports the note may be the same as or different from the hypermedia system. The content of the notebbox may be stored separately from the hypertext.

After entering comments into a notebbox, the box is closed and the user returns to normal activities. The notebbox is displayed when the user activates the notebbox link or icon by clicking on it with the mouse.

Notebox tools have traditionally had limited functionality as compared to the primary application. They rarely allow the user to create links to reference other notes or allow users to create notebboxes on notebboxes in a recursive-like manner. These limitations can be very confusing since users expect the same functionality to be available. It is highly recommended that the notebbox facility be an extension of the hypermedia document environment.

Another facility which is important to JSC is marginalia. A marginalia approach allows the user to write directly "on top of" documents, without invoking another tool or opening another window. Comments are normally placed in the margins of the document, but when necessary, can be placed directly on top of the information which is being displayed. This is especially valuable to support annotation of figures and drawings.

A marginalia approach employs a spatial interface metaphor whereby objects "stick" to the display wherever they are placed. This interface behavior is found in paint programs and is not characteristic of hypermedia environments. In fact, the only commercial system which supports this type of functionality is KMS [AMY88].

Both the notebbox and marginalia approaches are important and necessary in the JSC environment to support user annotation. Further, systems must assist in the management of annotations by keeping track of the author of the annotation and allowing users to select or filter annotations to be displayed, printed, or edited. Such a capability also will allow users to see annotations of other users, assuming appropriate permissions are provided.

Union and intersection operations are important to support the annotation space. For example, a user could request a display of annotations created by a specified set of users as compared to being limited to seeing either all or only one user's annotations. In this regard, color can be used to distinguish among the various users.

Link Markers

Link markers are the visual manifest of a link on the user's display. They provide the initiation point for activating a link and can act as the termination point at the link's destination.

When a link is followed and the destination is reached, color can be used to focus the user's attention to the object(s) which constitute the destination. Depending on the granularity of the destination, it may be appropriate to omit coloring the destination objects in a special way. For example, if the destination is a node, coloring the entire node may be very distracting. Instead, the node's border could be colored to indicate that the link connects to the entire node. If the destination is an object in a node, the individual object may be displayed in a different color to distinguish it from the other objects in the node. The choice of color is rarely a problem in this case unless the object is extremely large.

A final issue regarding color concerns the directionality of links and the status of the objects at the destination. If the link is bidirectional, the objects at the destination may be both destinations and sources of the same link. In the case of a uni-directional link, the object(s) may be the destination of one link and the source of a different link. The basic cases that must be delineated are listed below.

- not associated with a link
- source of a link
- destination of a link
- source of one link and destination of another link
- source and destination of multiple links

Identification

A fundamental issue in hypermedia concerns the ability to identify document components in order to relate the components through links. Users must be permitted to create links which connect to specific paragraphs, sentences, or phrases. Moreover, these relationships must be maintained when new versions of the same document are released. To provide this capability, a hypermedia system must identify objects in some manner. An identification mechanism, such as that proposed by Khoshafian and Copeland [KhC86], is appropriate for supporting hypermedia for different levels of object

granularity. When this model is applied to a document environment, it requires that a unique identifier be assigned to each object and used throughout the object's life. Implementing such a mechanism in the extremely large and diverse environment of NASA will be difficult due to the problem of controlling object identifiers. Thus, an alternative approach might be to use a combination (concatenation) of the document's identifier, JSC number, and a within-document identifier. This scheme appears to be sufficient for most needs except when the same object is used in multiple documents. In this case, the identity of the object must permit reference and containment in multiple documents. Additionally, links associated with the object must maintain the context of the link, that is, relative to the document in which the reference is made.

Another problem concerns document accessibility with respect to database issues. In many cases, it is highly desirable to provide access to objects without requiring the "opening" of an entire document which contains the object. Opening an entire document from which only a small part is needed results in overhead which is generally unnecessary. Thus, objects should be viewed as entities which may exist within or separate from their associated documents.

Finally, an object identification policy must provide the ability for crew and staff to alias or name objects using slang. Generally, object identification is viewed as a system level function and is of no concern to the end user. However, maintaining an aliasing facility which is unique to individual users is both costly in terms of document storage and configuration management. Regardless, aliased object identification has been expressed on several occasions and is viewed as a necessary part of the electronic document environment. The reader can consult the final report for DTO-1209 [NAS92] concerning formal requests from crew for aliasing services.

Links

The traditional view of linking is one of the user clicking with the mouse on an icon and either the current window's contents change to display new information or another window appears with the information. This example demonstrates a 1-1 relationship between objects but also demonstrates that the view provided to the user can be implemented differently. Either the current window's contents are replaced or a new window is displayed. Both mechanisms should be supported with the user being able to control which view is provided. Alternatively, the author of the link could specify the behavior during navigation of the link. For additional information concerning the importance of link arrival and departure, the reader should consult Slatin [Sla88].

Links may be associated with objects of various sizes including entire nodes, paragraphs, sentences, words, characters, figures, video clips, rules, sounds, and so on. Due to the diversity of the information that might be associated with links, links should support arbitrary object granularity. This would allow links to connect simple, complex, or node objects. For example, a word in a sentence could be linked to a video.

Multi-endpoint links are important and should be provided. This allows an author to create a link which connects two or more objects at the source end of the link with two or more objects at the destination of the link. The objects to which the link may be attached can be of the same or different types. This would allow the user to follow a link into a

document at which time a block of text, video clip, and sound byte appear and are played simultaneously. Such a scenario would be valuable for crew training or when information from multiple documents containing objects of different types must be accessed at the same time.

The user should be allowed to define links that can be initiated automatically or manually. The traditional view of hypermedia requires that the user initiate all navigation. This view is limiting in that it requires user intervention for all linking services. It is suggested that links be viewed as functional entities which support the document viewing environment. Links may be implemented to automatically "fire" at a certain time, under certain conditions, or when other links are navigated. A system which exhibits some of this functionality is described in [StF89].

HYPERMEDIA VIEWER

A prototypical hypermedia system to support the viewing of on-line documentation for the MIDAS [Pac92] system was constructed. The viewer supports full color text, graphic, and image objects on Unix, Sun³, workstation environments. The viewer is optimized for handling small documents and supports two "types" of documents -- a "roff-like" document construction language to facilitate document presentation and ASCII files.

There are many features of the system which are important to JSC and NASA. Some of the more notable features include spatial and notebbox annotation, full-color highlighting, simple and complex boolean search, catalog management with subclassing, user customization, and integration with existing systems. A supporting component provides users with the ability to parse and extract table of contents information from documents, creating a hypertext which provides non-linear access to the document. Another support component parses incoming electronic mail and builds a multi-level hypertext relative to the day and order that messages were received. The mail component embeds hypermedia buttons into messages to facilitate replies to the sender or to delete messages that are no longer valuable.

Four types of links are supported. The types delineate the placement, appearance, and functionality of the link. Links may be spatially placed *on* a document if identification of objects in the document is not possible. Links may be *embedded within* the document to provide linking services in the event that changes to the document affect the position of an object or if links are to "travel" with the document to support distribution. Links may be used to initiate applications, including the viewer itself. They can be packaged with search arguments to assist the user in locating information through a single mouse click. Automatic links allow the user to identify words or phrases which are defined across all documents and provide access to a document, perform a search, or initiate a process.

Links persist in both embedded and non-embedded forms and are managed by a hypermedia data management component. Users can create links which are maintained in a personal link space and are unavailable to other users. The linking mechanism also

³ Sun is a trademark of Sun Microsystems, Inc.

allows multiple users to share a common set of links over a document space.

The embedded link mechanism permits users to place links within documents so that links travel with and are maintained as part of the document content. In this case, the link is not associated with a particular user since user identification is not supported. The link is therefore a *public* entity and is accessible by all users of the document. One of the advantages of the embedded link mechanism is that it allows the creation of hypermedia documents by non-hypermedia components. For example, a background process could create a document containing embedded links and initiate the viewer to provide non-linear navigation over the document. Moreover, the process could run continually, affecting the content and structure of the hypertext dynamically. Such a process would not have to concern itself with interface, navigation, or search issues since all of these services are provided by the hypermedia viewer.

CHALLENGES AND UNTAPPED ISSUES

There are many challenges which face the use of hypermedia throughout NASA as well as other organizations. Some of these challenges are being addressed in current research while others remain virtually unexplored. In this section we present a few issues which have the potential of offering enhanced access or presentation features to the electronic environments at JSC, or which appear to present a significant challenge to the successful deployment of hypermedia throughout the Agency.

Cross-vendor compatibility

It is obvious that several systems must exist in order to support the various operations at NASA. Many of these systems utilize a proprietary format for storing information or exchanging information between components and are designed to operate on a unique computing platform. These limitations result in systems which are incompatible with others even though they may be implemented for the same computing platform. This lack of compatibility and interoperability is well documented in the literature and is regarded as a major limitation in hypermedia system design [Eng78, Eng82, Mey89, BoR90, Ri190].

Systems which exist as "islands" and which are unable and incapable of sharing information and procedures with other systems are inappropriate for the long-term goals of JSC. Since it is impossible for a single system to meet every need of JSC and NASA, multiple systems must exist within the framework of a larger environment. This requires vendors to cooperate and develop systems which are capable of interacting and exchanging information, especially in a sense much broader than the dynamic data exchange (DDE) services which some systems provide. This necessitates an overall plan of integration whereby systems coexist in a seamless manner to support ground and space operations.

An interoperable framework offers many benefits, one of which is the ability to add new components. Another benefit is the ability to migrate to new or updated systems which provide increased capabilities. Migration can be accomplished without requiring transformations of information or loss of functionality if systems utilize a compatible data

substrate. Achieving this level of interoperability will empower users by providing them with a collection of tools that can be used and tailored to meet specific needs. Therefore, an important task is to establish a framework for the interchange and sharing of information, and the integration of diverse systems within the NASA computing environment.

A related issue concerns a much broader view of the migration concept. This view is only now being considered within the scope of hypermedia systems but in fact has been an issue which has been addressed in other disciplines. The issue concerns the migration of processes to support computation.

The earlier paragraphs presented the issue of migration as one involving the movement of information, i.e. documents, around an information space. If one views the software components as "information", the migration issue equally applies. This perspective provides for the migration of processes among workstations to improve performance and distribute the workload. It also allows for recovery of a process if a workstation fails. To plan for the long-term needs of the Center, it is important to consider systems which utilize such architectures, allowing their components to be distributed and migrated without a negative effect on user activity.

Composition and Computation

The hypermedia research community has been exploring issues of *composition* and *computation* [Hal88] for quite some time. Composition is the construction of objects from other objects in the information space while computation involves dynamic, "on the fly", determination of object content and structure.

The incorporation of these capabilities into hypermedia systems appears to be very important to the immediate and long-term goals of electronic documentation at NASA for several reasons. First, hypermedia systems have traditionally employed a monolithic architecture and static data model [Mey89]. The monolithic (single process) architecture inhibits interoperability, as described in the previous section, and the static data model requires all links to be defined prior to the time of access. Clearly, the costs of manually defining every link is prohibitive given the enormous number of documents which exist in the NASA environment. Therefore, methods must be used to create links automatically and reliably on demand.

Second, users have different document access needs and for this reason it is important for a hypermedia system to construct links based on specifications from or information about a user. In order for this to be accomplished in a hypermedia framework, the content and the structure of objects must be determined on the fly.

Third, information can be a reusable resource in that it may be used in multiple documents. To support reusability in most existing hypermedia systems, information must be duplicated. Duplication is not cost effective given the tremendous number of documents which will exist in electronic form within the NASA community.

Systems which support referential and inclusion links and utilize a non-embedded hypermedia data model offer the greatest potential for supporting computability and

composition. Multiple link types based on reference and inclusion-style linking allow the user to construct hypertexts that can be used more effectively to tailor the presentation of information. Non-embedded links allow the system to determine link destinations without accessing the underlying documents. This allows the hypermedia engine to service the needs of multiple applications within a single computing environment. This type of design expands the scope of hypermedia by provides linking services "across" applications rather than limiting services to a single or selected class of applications. Further, this design allows the hypermedia engine to play an active role in support exploratory access from a structural perspective. For example, the hypermedia engine could construct and display a graphical map of the document space based on the current contents and relationships which exist among the documents. As another example, the engine could provide a list of all documents which contain a particular object and which is linked to a specific document and which consists of at least 4 chapters.

Finally, it is important not to confuse the issue of computation with that of automatic linking. Computation involves much more, specifically, the construction of the objects to be displayed as well as the construction of links that are associated with the object. In contrast, automatic linking can be implemented over a static data space and not involve any *computation* on the part of the hypermedia engine or application.

Distribution

The enormous number of documents and potential users necessitate an efficient and accurate distribution system throughout NASA. Unfortunately, no hypermedia system or research effort has addressed this issue. For this reason we identify this as an important issue and one that deserves special consideration.

There are various ways to think of document distribution which include the distribution of the underlying document, public and private links, and user-defined annotations. Obviously, the goal of a distribution system is to permit timely and easy access to documents. We identify and outline a few specific scenarios to demonstrate some of the possible responsibilities of a distribution system.

New document. A new document is released to the organization. The document has zero or more links. A list of target users, i.e. a distribution list, identifies a preliminary set of users to send the document and its links.

The potential problems concerning the distribution of a new document relate to the links which are associated with the document. More than likely, the links associated with this document will reference other documents in the information space. If the user is not on the distribution list for any one of the destination documents, the links which provide the reference are invalid for that user. This requires the hypermedia engine to coordinate document distribution and access lists with link visibility to avoid incorrect presentation of links when the document is viewed. Alternatively, links may serve as a specification for adding users to a document's distribution list and distributing the document to the user automatically.

Updated document. The release or versioning of a document incurs many of the same problems as those associated with a new document but also has the problem that any user-

defined links and annotations must be merged into the new release. Various mechanisms can be used to relate annotations and links with their intended positions, but depending on the degree of change to the document, some mechanisms may not adequately preserve these relationships. This usually results in dangling objects and links, or objects which must be discarded because their relationship to the new version cannot be determined. Clearly, the number of objects which must be discarded must be minimized in the NASA environment.

Cross platform and system. Users who conduct activities on multiple computing platforms or use multiple software systems risk inconsistency in versions of the same document. For example, if a user has access to a document on a personal computer and access to the "same" document on a workstation, the annotations and links defined by the user on their personal computer will probably not carry over to the workstation environment or vice versa, unless of course the two systems are networked and the document is shared between the systems or the user transports a copy of the document between the systems on a backup device (e.g. floppy disk). Clearly, requiring the user to manage documents and their distribution manually would involve significant time commitments for users given the number of documents to which they might refer. The networked solution is therefore more appropriate but requires the distribution system to have access to all computing platforms used by an individual and to maintain the annotations made to a document. This requires that all annotations and links be in the same format, a mapping operation occur to convert annotations made on one machine to a form which can be used on another, or all machines store and access annotations from a central repository.

Informal sharing. A final problem concerns the informal sharing of documents and annotations among users. In this case, a user shares a document with another user without the distribution system being informed of this activity. The new user proceeds to create annotations and links for this document in isolation and remains dependent on their friend for information about updates. This is problematic if the distribution system provides the mechanism for coordinating updates with user-defined annotations and links. Since the informal user is not on the distribution list, their annotations will not be merged into the new version. Although this may not be a problem for the first few releases of a document, it is probable that the informal user will eventually cease updating the versions with their annotations. Thus, the informal user eventually becomes completely out of sync with the release of the document due to their inability or unwillingness to maintain annotations and links.

SUMMARY

Electronic documents and systems are becoming an important means of managing information for ground and space operations at NASA. Hypertext and hypermedia technologies will play a strategic role in the management of information by allowing users to structure and access information by navigation.

This report has outlined some of the uses and problems facing the implementation and deployment of hypermedia systems at JSC. It also presents a brief description of a hypermedia system that was developed to support and demonstrate hypermedia capabilities for on-line documentation. Finally, the report discusses some hypermedia

technologies which have either been untapped by vendors or present significant challenges to the Agency.

REFERENCES

- [AMY88] R. Akscyn, D. McCracken, and E. Yoder, "KMS: A distributed hypermedia system for managing knowledge in organizations," *Commun. ACM* 31(7), 820-835 (July 1988).
- [BoR90] J. Bornstein and V. Riley, "Hypertext interchange format," *Proceedings of the Hypertext Standardization Workshop*, pp. 39-47 (January 1990).
- [Eng78] D. C. Engelbart, "Toward integrated, evolutionary office automation systems," *Proceedings of the International Engineering Management Conference*, pp. 63-68 (October 1978).
- [Eng82] D. C. Engelbart, "Integrated, evolutionary, office automation systems," pp. 298-315 in *Emerging Office Systems, Proceedings of the Stanford International Symposium on Office Automation.*, ed. R. Landau, J. Bair, and J. Siegman, Ablex Publications Corporation, Norwood, NJ (1982).
- [Hal88] F. Halasz, "Reflections on NoteCards: Seven issues for the next generation of hypermedia systems," *Commun. ACM* 31(7), 836-852 (July 1988).
- [KhC86] S. Khoshafian and G. Copeland, "Object identity," *Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications - OOPSLA '86, Sigplan Notices* 21(11), 406-416 (November 1986).
- [Mey89] N. Meyrowitz, "The missing link: Why we're all doing hypertext wrong," 107-114 in *The Society of Text: Hypertext, Hypermedia, and the Social Construction of Information*, ed. E. Barrett, The MIT Press (1989).
- [NAS92] NASA/JSC-DTO1209, "Development Test Objective (DTO) 1209 Advanced Portable Computer Evaluation - Final Report," NASA JSC Report #JSC-25588, Johnson Space Center, Houston, TX (December, 1992).
- [NAS93] NASA/JSC-EDP, "Electronic Documentation Project (EDP):Level A requirements," NASA JSC Report #JSC-TBD, Johnson Space Center, Houston, TX (June, 1993).
- [Nel65] T. H. Nelson, "A file structure for the complex, the changing, and the indeterminate," *Proceedings of the 20th National ACM Conference*, pp. 84-100 (1965).
- [Pac92] G. Pack, "Application of Automation and Intelligent Software to Analysis of Engineering Space Shuttle Systems - Mission Evaluation Room (MER) Intelligent Diagnostic and Analysis System (MIDAS)," 1992 JSC Research and Development Accomplishments Report (RADAR) Inputs Schedule, IA4, Chief Technologist, New Initiatives Office, Johnson Space Center, Houston, TX (1992).
- [Ril90] V. A. Riley, "An interchange format for hypertext systems: The intermedia model," *Proceedings of the Hypertext Standardization Workshop*, pp. 213-222 (January 1990).

- [Sla88] J. Slatin, "Toward a rhetoric for hypertext," *Hypermedia '88 Conference*, (August 1988).
- [StF89] P. D. Stotts and R. Furuta, "Petri-net based hypertext: Document structure with browsing semantics," *ACM Trans. Off. Inf. Syst.* 7(1), 3-29 (January 1989).